

Application No. : 09/242,383  
Amdt. Dated : January 15, 2004  
Reply To Office Action Of : August 15, 2003

**Amendments To The Specification**

Amendments to the specification herein show their changes in highlighted form, i.e., insertions appear as underlined text (e.g., insertions) while deletions appear as strikethrough text (e.g., ~~deletions~~). Thus, it should be understood that insertions should not actually appear in the specification as underlined text.

**In the Specification**

Please add the following new section heading on page 1, line 4:

(B) -- BACKGROUND --

Please replace the section heading at page 1, line 6 with the following rewritten section heading:

(B) -- ~~Field of the Invention~~ FIELD OF THE INVENTION --

Please replace the section heading at page 1, line 15 with the following rewritten section heading:

(B) -- ~~Description of the Prior Art~~ DESCRIPTION OF THE RELATED ART --

Please replace the section heading at page 4, line 24 with the following rewritten section heading:

(B) -- ~~Statement of the Invention~~ SUMMARY --

Please replace the section heading previously added in the Preliminary Amendment on page 8, line 17 with the following rewritten section heading:

(B) -- ~~Brief Description of the Drawings~~ BRIEF DESCRIPTION OF THE DRAWINGS --

Please replace the section heading at page 8, line 18 with the following rewritten section heading:

Application No. : 09/242,383  
Amdt. Dated : January 15, 2004  
Reply To Office Action Of : August 15, 2003

*B4* -- Detailed Description DETAILED DESCRIPTION --

Please replace the **bolded** text at page 12, lines 29 and 30 as follows:

*X*  
*B* -- **Pointer Descriptors, Buffer Descriptors and Heap Descriptors** Pointer Descriptors, Buffer Descriptors and Heap Descriptors --

Please replace the **bolded** text at page 13, line 1 as follows:

*B8* -- **Pointer Descriptors** Pointer Descriptors --

Please replace the **bolded** text at page 13, line 22 as follows:

*B9* -- **Buffer Descriptors** Buffer Descriptors --

Please replace the **bolded** text at page 14, line 14 as follows:

*B10* -- **Heap Descriptors** Heap Descriptors --

Please add the following new paragraphs after the paragraph ending on line 22 of page 14:

*B* -- TPtrC is a constant Descriptor through which no data can be modified. All of its member functions (except the constructors) are constant. TPtrC is shown schematically at Figure 1. TPtrC is useful for referencing constant strings or data; for example, accessing text built into ROM resident code, or passing a reference to data in RAM which must not be modified through that reference. TPtrC is derived from TDesC, which provides a large number of member functions for operating on its content; for example, locating characters within text or extracting portions of data.

TPtr is a modifiable pointer Descriptor through which data can be modified, provided that the data is not extended beyond the maximum length. The maximum length is set by the

Application No. : 09/242,383  
Amdt. Dated : January 15, 2004  
Reply To Office Action Of : August 15, 2003

constructor. TPtr points directly to an area in memory containing the data to be modified. TPtr is shown schematically in Figures 2A and 2B.

TBufC is a buffer Descriptor containing a length followed by the data area. Data can be set into the Descriptor at construction time or by the assignment operator (operator=) at any other time. Data already held by the Descriptor is constant. TBufC is shown schematically at Figure 3. The length of a TBufC is defined by an integer template; for example, TBufC<40> defines a TBufC which can contain up to 40 data items.

TBufC is derived from TDesC, which provides a large number of member functions for operating on its content; for example, locating characters- within text or extracting portions of data. TBufC provides the member function, Des( ), which creates a modifiable pointer Descriptor (a TPtr) to reference the TBufC. This allows the TBufC data to be changed through the TPtr, as indicated schematically at Figure 4. The maximum length of the TPtr is the value of the integer template parameter.

TBuf is a modifiable buffer Descriptor containing data which can be modified, provided that the data is not extended beyond its maximum length.. TBuf is shown schematically at Figure 5. The maximum number of data items that the data area within TBuf can contain, is defined by the maximum length. The length of the Descriptor indicates how many data items are currently contained within the data area. When this value is less than the maximum, a portion of the data area is unused. The maximum length of a TBuf is defined by an integer template; for example, TBuf<40> defines a TBuf which can contain up to data items (and no more!). A TBuf is useful for containing data which needs to be manipulated and changed but whose length will not exceed a known maximum; for example, word processor text. TBuf is derived from TDes which, in turn, is derived from TDesC. Therefore, it inherits all the const member functions defined in TDesC plus the member functions from TDes which can manipulate and change the data; for example, appending a character to the end of existing text.

HBufC is a Descriptor containing a length followed by data. It is allocated on the heap using the New( ), NewL( ) or NewLC( ) static member functions. The length of the Descriptor is passed as a parameter to these static functions. HBufC is shown schematically at Figure 6. Data can be set into the Descriptor at construction time or by the assignment operator (operator=) at any other time. Data already contained by the Descriptor is constant. HBufC is derived from

Application No. : 09/242,383  
Amdt. Dated : January 15, 2004  
Reply To Office Action Of : August 15, 2003

TDesC, which provides a large number of member functions for operating on its content; for example, locating characters within text or extracting portions of data.

HBufC provides the member function, Des( ), which creates a modifiable pointer Descriptor (a TPtr) to reference the HBufC. This allows the HBufCC data to be changed through the TPtr. The maximum length of the TPtr is the length of the HBufC data area. Figure 7 illustrates this schematically.

All of the Descriptor classes TPtrC, TPtr., TBufC, TBuf and HBufC are derived from the abstract base classes TDesC and TDes. The class TBufCBase, although marked as an abstract class, is merely an implementation convenience. Figure 8 schematically illustrates the relationship between the classes.

#### EXAMPLE FUNCTIONS (See Appendix 1 for details and additional functions)

The code fragments illustrate the use of Left().

```
...
TBufC<8>str(_L("abcdefg"));
...           // returns a TPtrC descriptor
str.Left(4); / representing the sting
...//          "abcd"
```

The result of this specific example can be visualized in a before (shown in Figure 9) and after (shown in Figure 10) fashion. The underlined text in the “after” diagram (Figure 10) indicates the data represented by the returned descriptor.

The code fragments illustrate the use of Right().

```
...
TBufC<8> str(_L("abcdefg"));
...           // returns a TPtrC descriptor
str.Right(4); // representing the string
...           // "defg"
```

The result of this specific example can be visualized in a before (Figure 11) and after (Figure 12) fashion. The underlined text in the “after” diagram (Figure 12) indicates the data represented by the returned descriptor.

Application No. : 09/242,383  
Amdt. Dated : January 15, 2004  
Reply To Office Action Of : August 15, 2003

The code fragments illustrate the use of AllocL( ).

```
...
TBufC<16> str(_L("abcdefg"));
HBufC* ptr;
...
ptr = str.AllocL(); // Returns address of new HBufC descriptor
... // holding the string "abcdefg".
ptr.Length(); // Returns the length 7
...
```

The result of this specific example can be visualised in a before (Figure 13) and after (Figure 14) fashion.

The following code fragments illustrate the use of Justify().

```
...
TBuf<16> tgt(_L("abc"));
...
tgt.JustifyLL("xyz"),8,ECenter,'@');
```

The descriptor tgt has a maximum length of 16 and initially holds the string “abc”. After the call to Justify(), the content of tgt changes to “@@@xyz@@@” as illustrated at Figure 15.

In this example, the content of the source descriptor is taken to form an 8 character field which replaces the original content of the descriptor tgt. The characters “xyz” are centred within the new field and padded on both sides with the fill character’@’. Setting the alignment to ELeft would change the content of tgt to “xyz@@@@@@” while setting the alignment to ERight would change the content of tgt to “@@@@@xyz” In all three cases, the length of the descriptor tgt changes from 3 to 8.

```
...
TBuf<8> tgtLL("abc"); ...
...
tgt.JustifyLL("xyz"),9,ECenter,'@');
```

This call to Justify() will panic because the resulting length of data in tgt would exceed the maximum length of tgt.

```
...
TBuf<16> tgt(_L("abc"));
...
tgt.Justify(_L("rstuvwxyz"),8,ECenter,'@');
```

Application No. : 09/242,383  
Amdt. Dated : January 15, 2004  
Reply To Office Action Of : August 15, 2003

In this call to Justify(), the content of tgt changes to "rstuvwxyz" as illustrated at Figure 16. Only eight of the nine characters in the source descriptor's data area are copied.

The following code fragments illustrate the use of AppendJustify().

...  
TBuf<16> tgt(\_L("abc"));  
tgt.AppendJustify(\_L("xyz"),8,ECenter,'@');

The descriptor tgt has a maximum length of 16 and initially holds the string "abc". After the call to AppendJustify(), the content of tgt changes to "abc@@xyz@@@@" as illustrated at Figure 17.

The following code fragments illustrate the use of AppendJustify().

...  
TBuf<16> tgt(\_L("abc"));  
tgt.AppendJustify(\_L("xyz01234456'789"),3,8,ECenter,'@');

The descriptor tgt has a maximum length of 16 and initially holds the string "abc". After the call to AppendJustify(), the content of tgt changes to "abc@@xyz@@@@" as illustrated at Figure 18.

The following code fragment depicted in Figure 19 illustrates the use of ZeroTerminate().

...  
TBuf<8> tgt(\_L("abcde"));  
...  
tgt.ZeroTerminate()  
...

The length of the descriptor tgt is 5 both before and after the call to ZeroTerminate(). --

Please delete pages 15 to 165 (these pages are being moved to an appendix and will continue to form part of the patent application).

Please replace the section heading at page 169, line 2 with the following rewritten section heading:

-- Abstract ABSTRACT --

**Application No. : 09/242,383**  
**Amdt. Dated : January 15, 2004**  
**Reply To Office Action Of : August 15, 2003**

Please add Appendix 1 (attached to this amendment) to the end of the specification.